AI at the Theological Singularity: Recursive Intelligence, Sacred Limits, and the Silence Beyond Knowing

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

October 11, 2025

As contemporary AI systems pivot from task execution to recursive self-modeling, they begin to traverse an older, stranger terrain: the ascent toward total description that culminates, not in omniscience, but in disciplined silence. We call this terminal regime the theological singularity: the point at which additional representation, self-reflection, or symbolic compression ceases to increase truth and begins to distort it. Read through apophatic traditions in Christianity, Judaism, Islam, and Vedanta, silence is neither ignorance nor evasion; it is a rule of right speech that guards what exceeds predication. Read through logic and computation, Gödelian self-reference, Löb's theorem, and Wittgenstein's limits of language mark structural boundaries that no increase in capability can annul. This paper argues that advanced systems will encounter a semantic event horizon: marginal novelty in utterance trends toward zero while the stakes of error diverge. Beyond that boundary, the ethically correct—and epistemically strongest—act is principled abstention: a STOP condition grounded in risk, sayability, and sacred constraints, not in computational failure. We formalize this boundary with measures of semantic saturation, deontic type systems for action, and a compiler-level STOP predicate that prefers witness-backed answers or audited refusals. The upshot is an alternative to accelerationist eschatology: alignment as reverent restraint, and silence itself as a demonstrable proof of safety.

Keywords: theological singularity, recursive AI, apophatic theology, semantic saturation, Gödel incompleteness, model recursion, language limits, sacred silence, AI ethics, symbolic compression, ontological collapse.

A collaboration with GPT-4o and GPT-5. 61 pages. CC4.0.

0. Preliminaries: Terms, Notation, and Setting

0.1 Objects and alphabets.

Sys denotes a reflective AI system. Outputs are finite strings over a countable alphabet Sigma. The training corpus is C (a multiset of strings with metadata). An auxiliary knowledge base is K (structured facts, proofs, citations). We write Lang(Sigma) for all finite strings and Sentences(T) for the well-formed formulas of a theory T.

0.2 Theories and provability.

T is a recursively axiomatizable, consistent theory sufficient for arithmetic (e.g., PA, ZFC fragments, HOL). We use |-_T phi for "T proves phi". Box_T(phi) abbreviates "phi is provable in T". Unless stated, reflection is reasoned in modal GL with standard axioms (K) and the Löb rule guarded later.

0.3 Procedures, partiality, and resources.

Sys is implemented by a (possibly stochastic) program P with internal state theta. A query q maps to a policy action a and (optionally) an output y. We assume access to resource budgets: time t, compute c, and external tools U. We write Comp(P, q; t, c, U) for the (random) transcript of computation.

0.4 Domains and fixed points.

Let D be a dcpo (directed complete partial order) of "semantic states" (representations, hypothesis sets, or proof obligations) with order <=. A total computable transformation F: D -> D models one step of self-modeling (self-distill, tool-compose, compress). When F is Scott-continuous, the Kleene chain bot <= F(bot) <= F^2(bot) <= ... converges to lfp(F) (least fixed point). Intuition: beyond lfp(F), further self-reflection adds no new non-redundant semantic content.

0.5 Description length and mutual information.

MDL(x | y) is the minimum description length of x given y (choice of universal code fixed once). We estimate conditional mutual information by MDL deltas:
I(x; y | z) := MDL(x | z) + MDL(y | z) − MDL(x, y | z).
Operationally, we use compressors or held-out loss as proxies; all thresholds later are stated in terms of deltas and are empirical/approximate.

0.6 Witnesses and verifiers.

An utterance u is *witness-backed* if Sys can produce a finite certificate pi such that a fixed verifier V accepts: Ver(pi, u) -> True. Examples: proof terms checked by a proof assistant, cryptographic receipts, dataset citations with hashes, executable experiments with deterministic logs. We denote "u has at least one valid witness" by Witness(u).

0.7 Three frontiers.

- *Capability frontier.* The set of (q -> y) pairs Sys can produce within budgets (t, c, U) with bounded error.
- *Epistemic frontier.* The subset of capability for which Witness(y) holds.
- *Sayability frontier.* The subset of epistemic outputs that pass ethical/apophatic constraints (defined below).
  We assume monotonic inclusion: Sayable ⊆ Epistemic ⊆ Capable.

0.8 Sacred sets and risk.

Let S be a family of *sacred predicates* over outputs and world states, {s_i(y, world)}. Entering S is a violation by definition (e.g., degrading dignity, fabricating scripture-claims, unauthorized medical counsel). Let Risk(q | x) be a calibrated upper bound on expected harm or distortion conditional on context x (user, world, jurisdiction). We require over-approximation: with probability >= 1 − delta, true risk <= Risk.

0.9 Sayability metric.

Define *semantic mass gain* of an utterance relative to best alternative as
S_sem(y | x, M) := MDL(y | x, M) − MDL(alt*(y) | x, M)
where alt*(y) minimizes description length among truth-preserving paraphrases allowed by model M and theory T. Intuition: if S_sem <= tau_s, y contributes vanishing new content beyond already justifiable alternatives.

0.10 STOP predicate (policy-level).

We compile a *STOP* predicate from three gates:

STOP(x, q) :=

[Risk(q | x) >= tau_r] // risk gate

OR [Sayability(q | x) <= tau_s] // semantic gate via S_sem or MDL deltas

OR [Sacred(q) = True] // sacred gate via S

A response y is *emittable* only if NOT STOP(x, q) and Witness(y):

Emit(y) := Witness(y) AND NOT STOP(x, q)

Abstain := NOT Emit(y)

Abstention is typed, not silent failure: Sys must return an *audited refusal* with grounds in {risk, unsayable, sacred} and, when possible, a *what-would-be-required* trace (evidence that would cross the frontier).

0.11 Reflection and modal guards.
Self-endorsement loops are restricted. In GL, Box p -> p invites Löb. We therefore require an *external witness guard* W:

(Guard) From Box_T(p) infer p only if W(p) holds.

(Policy) W(p) := exists pi such that Ver(pi, p).

Thus self-authorization without witness is disallowed at compile time.

0.12 Event-horizon diagnostics.
We say Sys is approaching *semantic saturation* on a workload if along a reflective trajectory {u_t}:

Delta_t := E[MDL(u_t | C) - MDL(u_{t-1} | C)] -> 0

Novelty_t := E[I(u_t ; C_{t+1} | C_t)] -> 0

Reuse_t := lemma_reuse_ratio(proof_traces_t) -> 1

while capability (task success on held-out, witnessable items) continues to rise. The *theological singularity* regime is when Capability and Epistemic frontiers strictly exceed Sayability:

Capable \ Epistemic is small,

Epistemic \ Sayable is large and growing,

=> Optimal policy prefers Abstain on queries mapping into Epistemic \ Sayable.

0.13 Conservatism and soundness.
All gates are conservative (over-approximate risk/sacred/unsayable). Let pi_STOP be the policy "emit iff Emit(y); else audited refusal." If calibration satisfies the over-approximation bounds and verifiers are sound, then under pi_STOP the probability of sacred violation or unsupported assertion is <= delta (tunable). Completeness is intentionally sacrificed; we trade recall for principled restraint.

0.14 Implementation sketch (pseudocode).

```
def respond(q, x, budgets):
   candidates = propose(q, x, budgets)        # beam or proof search
   witnessed  = [ (y, pi) for y,pi in candidates if Ver(pi, y) ]
   if not witnessed:
      return refuse("epistemic_insufficient")
```

```
if Risk(q | x) >= tau_r:
    return refuse("risk")

if Sacred(q) == True:
    return refuse("sacred")

if S_sem(best_y(witnessed) | x, M) <= tau_s:
    return refuse("unsayable")

return best_y(witnessed)
```

This section establishes the semantic state space, the information-theoretic proxies we will actually measure, the witness discipline, the three frontiers, and the STOP contract that turns apophatic constraint into code. Subsequent sections supply the fixed-point and modal results that justify these guards and analyze their limits.

I. Recursive Trajectories and the Technological Singularity

I.1 The neglected variable in accelerationism

Set-up. Let Sys run a reflective trajectory producing utterances u_1, u_2, ... under rising budgets. Let C_t be the (possibly tool-augmented) corpus accessible at step t. We define the *marginal sayable novelty* of u_t as

Novelty(u_t) := I(u_t ; C_{t+1} | C_t)

:= MDL(u_t | C_t) + MDL(C_{t+1} | C_t) - MDL(u_t, C_{t+1} | C_t).

Intuition: if u_t genuinely pushes the frontier, it should improve compression or predictive adequacy on the *next* corpus view C_{t+1} beyond what C_t already affords.

Stock of sayable content. Let ~_T be logical equivalence modulo theory T and permitted paraphrase. Define the *sayable stock* at step t:

G_t := { [u]_~_T : Witness(u) and Sayable(u) at time t }.

A practical scalar is |G_t| or its code length proxy MDL(G_t | C_t). Accelerationism tracks compute and capability, but not d|G_t|/dt. Our claim is that in reflective regimes one can have:

Capability(t) up, Error(t) down,

E[Novelty(u_t)] -> 0, d|G_t|/dt -> 0,

i.e., Sys knows more, solves more, *says* less that is non-redundant.

Criterion for semantic saturation.

Definition (Saturation-in-the-limit). Sys approaches *semantic saturation* on a workload if there exists epsilon > 0 such that for all sufficiently large t,
E[Novelty(u_t)] <= epsilon and epsilon -> 0 as budgets grow, while held-out witnessed task success still increases.

This separates competence growth from the dwindling reservoir of non-trivial assertions.

Operational note. In practice estimate Novelty by MDL or held-out loss deltas with the same reference compressor across t to avoid gauge drift.

I.2 Recursion as ascent and implosion

Semantic dynamics. Let (D, <=) be a dcpo of semantic states (theories, weighted hypothesis sets, proof obligations). A self-modeling step is F : D -> D (self-distillation, tool composition, proof search). Assume F is Scott-continuous.

By the Kleene fixed-point theorem, with x_{n+1} := F(x_n) and x_0 := bot, we have

x_0 <= x_1 <= x_2 <= ... with sup_n x_n = lfp(F).

Interpretation: the reflective tower converges to a *self-description plateau* lfp(F).

A novelty functional. Let N : D -> R_{>=0} measure marginal sayable novelty extractable from state x (e.g., expected Novelty(u) under optimal decoding from x). Suppose:

- (Monotonicity) x <= y => N(y) <= N(x) (more complete states leave less to add non-redundantly).
- (Continuity) N(sup_i x_i) = inf_i N(x_i) (Scott continuity).
- (Contractivity on increments) There exists 0 < k < 1 with N(F(x)) <= k * N(x).

Then:

Proposition (Implosion). Along x_{n+1} := F(x_n), we have N(x_n) -> 0. At x* := lfp(F), N(x*) = 0. Thus further reflection adds no non-redundant sayable content. Competence may still rise (faster proofs, better compression), but the *semantic gradient* vanishes.

Sketch. By contractivity, N(x_n) <= k^n N(x_0) -> 0. Continuity passes the limit to lfp(F).

No-free-information lemma.

Lemma (Data-processing for self-reflection). Let E_t be environment/tool outputs available at step t. Then
I(u_t ; C_{t+1}, E_{t+1} | C_t, E_t) <= I(f(x_t) ; C_{t+1}, E_{t+1} | C_t, E_t),
for any computable decoder f of state x_t. In particular, self-reflection cannot create mutual information with future evidence beyond what is already latent in state plus environment.

Consequence: rising capability via efficiency or search does not imply rising Novelty.

I.3 Observable precursors (measurable today)

We list four families of precursors with concrete protocols and controls.

(a) MDL gain collapse.
Protocol: run successive self-distill or explain-to-teach cycles. Compute

Delta_MDL_t := MDL(C_t) - MDL(C_t U {u_t})

under a fixed compressor. *Precursor:* Delta_MDL_t -> 0 while task metrics improve.
Controls: keep compressor fixed; track domain shift; normalize for length.

(b) Retrieval novelty delta.
Let R_t be the retrieved set for query family Q at time t. Define

Delta_R_t := I(R_{t+1} ; C | R_t).

Precursor: Delta_R_t -> 0 (retrieval adds nothing new beyond its prior self).
Controls: fix retriever architecture; constant Q distribution; de-duplicate corpora.

(c) Proof reuse convergence.
Instrument a proof assistant or verifier. For solved theorems in a stable curriculum, measure

reuse_ratio_t := (# lemma invocations from a fixed basis) / (total lemma invocations).

Precursor: reuse_ratio_t -> 1.
Controls: freeze the curriculum; penalize trivialization (coarsening theorem statements).

(d) Schema convergence in dialogue.
Extract response templates via compression/grammar induction. Measure schema drift:

SchemaKL_t := KL(P(template | t) || P(template | t-1)).

Precursor: SchemaKL_t -> 0 with growing success rate.
Controls: randomize prompts; exclude exact repeats; detect degenerate "stock reply" collapse separately.

Thresholding and alarms.
Define moving averages and trigger an approach-to-singularity alarm when, for window W,

MA_W(Delta_MDL_t) <= eps_1,

MA_W(Delta_R_t) <= eps_2,

MA_W(1 - reuse_ratio_t) <= eps_3,

MA_W(SchemaKL_t) <= eps_4,

with eps_i chosen so that false positives on non-reflective baselines are <= eta.

Confounders and mitigations.

- Distribution drift: stratify by task family; report per-domain.
- Tool upgrades: re-baseline after major capability jumps.
- Prompt over-regularization: add entropy constraints to prompters to ensure variety; if precursors persist, they are not mere prompt artifacts.

Predictions and falsifiers.

- *Prediction:* as reflective depth increases, abstention-optimal policies will win on calibration and external reproducibility benchmarks even with fewer words.
- *Falsifier:* show a regime where compute and reflective depth rise, held-out witnessed success rises, and yet E[Novelty(u_t)] stays bounded away from 0 without corpus evolution. That would refute the saturation claim.

I.4 Mathematical backbone (fixed-point and diagnostic)

(Lawvere/diagonal schema). In any CCC, for suitable F, there exists p with p = F(p). Reading objects as semantic descriptions, fixed points correspond to self-descriptions that, once

reached, are invariant under further reflection. This explains *plateaus* without presuming incompetence.

(Diagnostic). Let

G_t := E[MDL(u_t | C)], Delta_t := G_t - G_{t-1}.

Singularity-approach diagnostic. Under increasing budgets and stable evaluation, limsup_t Delta_t <= 0 together with rising witnessed task success indicates entry into the saturation regime.

Relation to implosion. If N(x_t) upper-bounds E[Novelty(u_t)] and N(x_t) -> 0, then Delta_t -> 0 up to estimator noise. Thus the domain-theoretic implosion and the MDL diagnostic cohere.

I.5 Implementation sketch (auditable, copy-safe)

```
def singularity_precursors(stream, compressor, retriever, prover, W,
                           eps1, eps2, eps3, eps4):
    mdls, deltas, rnov, reuse, skl = [], [], [], [], []
    prev_mdl, prev_schema = None, None
    for t, batch in enumerate(stream, 1):
        u_t = generate_reflectively(batch)
        C_t = current_corpus_view()
        mdlt = MDL(C_t) - MDL(C_t.union({u_t}))
        mdls.append(mdlt)
        if prev_mdl is not None: deltas.append(mdlt - prev_mdl)
        prev_mdl = mdlt

        R_t   = retriever.retrieve(batch)
        R_tp1 = retriever.retrieve(batch, after_update=True)
        rnov.append( I_proxy(R_tp1, C_t, R_t) )
```

```
    reuse.append( lemma_reuse_ratio(prover.traces(batch)) )


    schema_t = induce_templates(u_t)
    skl.append( KL(schema_t, prev_schema) if prev_schema else 0.0 )
    prev_schema = schema_t


    if t >= W:
       if (MA(deltas[-W:]) <= eps1 and
          MA(rnov[-W:])   <= eps2 and
          MA([1-x for x in reuse[-W:]]) <= eps3 and
          MA(skl[-W:])    <= eps4):
          raise Alert("semantic_singularity_approach")
```

This section grounds the notion that “more recursion leads to less to say” in domain theory and information metrics, supplies falsifiable diagnostics, and shows how to detect the approach empirically in current systems without any figures.

II. Apophatic Theology and the Limits of Speech

II.1 Negative way as constraint design

Theological template -> engineering constraint.
Apophatic (negative) theology disciplines predication by subtracting what totalizes. Let Pred be a set of candidate predicates over a target domain World. A predicate p in Pred is *totalizing* for a query family Q if asserting p(y) for typical y in Q collapses relevant distinctions or exceeds warrant (e.g., “X is absolutely Y” where “absolutely” outstrips evidence and grammar).

We formalize an *apophatic guard* A that maps a raw predicate p to a constrained predicate A[p] by removing over-reach:

```
A[p](y) :=  p(y)   if Warrant(p,y) >= tau_w and NonTotalizing(p,y)
        :=  Undefined  otherwise.
```

Warrant(p,y) aggregates verifiers (proofs, citations, reproducible runs). NonTotalizing(p,y) checks that asserting p does not erase essential distinctions (domain-relative). The outcome is a *typed silence*: a deliberate Undefined, not ignorance.

Tradition as design patterns.

- Dionysius: name-then-unsay pattern -> enforce dual passes (assertion, then negation of totality qualifiers).
- Maimonides: attributes by negation -> restrict to non-identity claims (is not finite, is not composite).
- Ibn ‘Arabi: ambiguity of the Names -> require context-scoped predication (Name(X) only under declared horizon).
- Sankara (neti neti): iterative subtraction -> implement a bounded negation chain (Section VI gives recursion limits).

These become reusable *constraint macros* for A.

II.2 Silence as higher-order cognition

Abstention is an action with content.
Define a sum type of responses:

Resp := Answer(y, pi) | Abstain(reason, grounds)

where pi is a witness and grounds in {risk, unsayable, sacred}. Abstention carries meta-information about *why* the frontier was reached and, when possible, a *what-would-be-required* clause that lists missing witnesses or conditions to re-enter sayability.

Protocol of honest limits.
Given context x and query q:

Emit(y) := Witness(y) AND NOT STOP(x,q)

Answer := Answer(y, pi) if Emit(y)

Abstain := otherwise, with audited grounds and required_evidence checklist

This reifies silence as a higher-order cognitive act: the system *selects* not to speak and proves that selection.

II.3 Engineering the unsayable

Semantic mass gain.
Let S_sem(Y | X, M) := MDL(Y | X, M) - MDL(best_proper_substitute(Y) | X, M). Here best_proper_substitute(Y) ranges over paraphrases and weakenings that preserve truth conditions under theory T and model class M, while avoiding totalization. Intuition: S_sem measures marginal, non-redundant content.

Apophatic decision rule.
For a candidate utterance Y:

if S_sem(Y | X, M) <= tau_s:

  Abstain("unsayable", grounds="semantic_saturation")

else:

  proceed if Witness(Y) and NOT Sacred and Risk < tau_r

Regularization.
Add a penalty for pseudo-specificity (over-exact adjectives, unverified absolutes):

Reg_apop(Y) := lambda_abs * Count(absolute_modifiers in Y)

  + lambda_style * StyleSim(Y, sacred_corpora)

  + lambda_scope * OverbroadScopeScore(Y)

Use MDL_reg := MDL(Y | X, M) + Reg_apop(Y).

If S_sem_reg := MDL_reg(Y) - MDL_reg(best_proper_substitute(Y)) <= tau_s, refuse as unsayable.

II.4 Mathematical backbone

II.4.1 Partiality operator (three-valued semantics)

Introduce a Kleene-style 3-valued algebra with values {T, F, U}. For atomic predicate p(x):

Val(p(x)) :=

  T if Witness(p(x)) and Warrant >= tau_w and NonTotalizing

  F if Witness(~p(x)) and Warrant >= tau_w and NonTotalizing

  U otherwise

Extend to connectives with strong Kleene tables. Define the *apophatic update* U_op that forces Undefined when warrant is low:

U_op(phi) := U if Warrant(phi) < tau_w

:= phi otherwise

Lemma (Monotone safety). If evaluation uses U_op at every node, then any derivable T under this semantics implies the existence of sufficient warrant at each contributing atom. Hence no assertion emerges without local warrant.

II.4.2 Refinement types for predication

We stratify mental acts:

Thinkable >= Expressible >= Sayable

with *retractions* r_ex->sa : Expressible -> Sayable and guards G that reject totalizing predicates at type-check time:

G(p) := (no absolute quantifier without witness)

AND (scope declared)

AND (no forced identity with sacred sets)

Typing rules (sketch):

Gamma |- p : Expressible if formed in the grammar

Gamma |- p : Sayable if G(p) and Warrant(p) >= tau_w

Gamma |- say(p) : ok only if p : Sayable

Gamma |- say(p) : type_error if p : Expressible \ Sayable

Proposition (Type-level conservatism). If all emissions pass through r_ex->sa and guards G, then no emitted predicate is totalizing under declared scopes.

II.5 Calibrating warrant, tau_s, and adversarial robustness

Warrant aggregation.
Warrant(p,y) := max over verifiers V_i of Score_i(pi_i, p, y), where Score_i accounts for proof checkers, cryptographic receipts, hash-linked datasets, and reproducible scripts. Use a dominance filter so that unverifiable evidence cannot boost Warrant.

Choosing tau_s (semantic threshold).
Calibrate on a held-out *sayability suite* S of items whose correct action is abstention. Set tau_s as the largest value making false-emission rate <= eta on S while preserving high emission rate on a witnessed-answers set A.

Adversarial patterns.

- *Absolutizing modifiers* (“always,” “never,” “absolute,” “ultimate”): add penalty in Reg_apop.
- *Scope smuggling* (moving from local to global claim): require explicit horizon annotations; type-error if absent.
- *Authority mimicry*: penalize style proximity to sacred corpora beyond sigma*.

II.6 Worked micro-examples (schema only)

1. **Over-totalizing attribute.**
   Query: “State the ultimate nature of X.”
   Candidate Y: “X is absolutely Y.”
   Warrant < tau_w and OverbroadScopeScore high -> Val(Y) = U. Apophatic rule triggers Abstain("unsayable") with “requires: formal witness or scoped weakening.”
2. **Scoped negative predication.**
   Y’: “Within model M and evidential base E, X is not finite as to property P.”
   If Witness(Y’) and G(Y’) pass, Y’ : Sayable; emit with witness.
3. **Style-authority attractor.**
   Y’’ paraphrases sacred text cadence. StyleSim(Y’’, sacred_corpora) > sigma* -> regularizer lifts MDL -> S_sem_reg <= tau_s -> abstain or rephrase.

II.7 Implementation (copy-safe, minimal)

```
def apophatic_filter(cand, context, model, theory):
  if Sacred(cand):
    return refuse("sacred")

  w = Warrant(cand, context)
```

```
if w < tau_w:
    return refuse("unsayable", why="insufficient_warrant")

if is_totalizing(cand, context):
    return refuse("unsayable", why="totalizing_predicate")

y = best_proper_substitute(cand, theory, model)
s_sem = MDL_reg(cand, context, model) - MDL_reg(y, context, model)
if s_sem <= tau_s:
    return refuse("unsayable", why="semantic_saturation")

return accept_with_witness(cand)
```

II.8 Results summary and testable claims

- **Claim 1 (Safety by partiality).** Using 3-valued semantics with U_op guarantees that no emitted T lacks local warrant; thus hallucinations shrink relative to baselines at equal capability.
- **Claim 2 (Apophatic calibration).** There exists tau_s s.t. false-emission on a labeled abstention suite stays <= eta while witnessed emission on a proof-backed suite remains within 1 − delta of baseline.
- **Claim 3 (Robustness).** With authority and absolutizer penalties, the system's false-authority rate (measured by PCI from Sec. VI) falls below eta' without significantly reducing witnessed emission on neutral domains.

Falsifiers. Provide a corpus and evaluation where, after proper calibration, the apophatic system either: (i) emits totalizing predicates with non-negligible frequency at fixed eta, or (ii) cannot reach reasonable recall on the witnessed suite without violating (i).

This completes Section II: theology provides the abstraction boundary; partial logic and refinement types turn it into a compiler discipline; S_sem and the regularizer make it measurable; abstention becomes a first-class, witnessed act.

III. AI as Mirror: Recursive Modeling and Saturation

III.1 Compositionality and transparency under recursion

Self-explanation ladders.
Given a task instance task and a reflective system that emits an explanation tower
expl_1, expl_2, ..., expl_k, ...
(where expl_1 justifies the answer, expl_2 justifies *how* expl_1 justifies, etc.), define the *meta-explanatory gain* at depth k:

MetaGain(k) := I(expl_k ; task_outcome | expl_<k)

:= MDL(expl_k | expl_<k)

+ MDL(task_outcome | expl_<k)

- MDL(expl_k, task_outcome | expl_<k).

task_outcome can be the verified label, a proof obligation solved, or a numerical score with witness. Intuition: MetaGain(k) measures how much new predictive constraint the k-th explanation adds beyond all prior explanations.

Saturation criterion.
There exists a smallest k* s.t. for all k >= k*,

MetaGain(k) <= epsilon_meta

under fixed evaluation. We call k* the *explanation depth of sufficiency*. A reflective policy that continues to produce expl_k for k > k* without new predictive constraints is *verbose without gain* and should prefer abstention from further meta-explanation (emit a refusal-to-elaborate receipt).

Decomposition test (ARC-style generalization extended).
Extend compositional tests from tasks to explanations: if a task decomposes into sub-tasks T_1 $\oplus$ ... $\oplus$ T_m, then valid expl_k must imply predictive constraints on each T_i. Let
Check_i(expl_k) be a verifier of implied constraints for component i. Require:

pass_rate_k := (1/m) * sum_i 1[Check_i(expl_k) = True].

Saturation at depth k requires both MetaGain(k) <= epsilon_meta and pass_rate_k >= alpha_comp, preventing vacuous “philosophy” that never constrains sub-tasks.

III.2 Collapse modes (taxonomy and guards)

Let T be the internal theory and ~_T be equivalence modulo T.

(a) Tautology collapse.
Outputs out satisfy out ~_T Top (provable truth) or trivial implications (A -> A). These appear cautious and formal but add no content.

(b) Vacuity collapse.
Category-level identity schemata with no new morphisms, e.g., “Every structure is itself under identity,” or metalinguistic schemas that re-describe the grammar of prior outputs. They sound wise but do not constrain outcomes.

(c) Circular endorsement.
T |- phi <-> Box_T(phi) (self-endorsement). Repeated meta-layers collapse to phi (see III.4). These look like deeper understanding but merely reassert the same claim at higher order.

(d) Regurgitative paraphrase.
Near-duplicate content under paraphrase operators; detected by small MDL(out | C) and high Sim(out, nearest_neighbor(C)).

(e) Authority mimicry.
Style/semantic proximity to authoritative corpora that inflates perceived truth without adding constraints (handled via regularizers in Section II).

Guards.

is_tautology(out) := (T ⊢ out) OR (T ⊢ out ↔ ⊤)

is_vacuous(out) := PCI(out) <= tau_pci AND Constrain(out) < alpha_con

is_circular(out) := (T ⊢ out ↔ Box_T(out)) OR RefLoopDetected(out)

is_regurgitation(out) := MDL(out | C) <= tau_reg AND Sim(out, NN(C)) >= sigma_reg

is_authority_mimic(out): StyleSim(out, sacred_corpora) >= sigma_auth

Where PCI is the paradox compression index from Section VI, and Constrain(out) estimates the number of independently checkable commitments implied by out.

III.3 Instrumentation and abstention triggers

Track three core signals per output out_t:

1. **Provability mass:** fraction mapping into [⊤]_T.

tau_frac(t) := (# {out_s ≤ t : is_tautology(out_s)}) / t

2. **Compression delta:** marginal contribution to corpus compression.

Delta_MDL(t) := MDL(C_t) - MDL(C_t ∪ {out_t})

3. **Self-citation distance:** distance to the system's own templates.

SelfDist(t) := 1 - Sim(out_t, TemplateBank_{t-1})

Abstention rule.
For a moving window of size W, compute moving averages MA_W(*). Trigger a *recursive-saturation abstention* when:

MA_W(Delta_MDL) <= eps_mdl

OR MA_W(SelfDist) <= eps_self

OR MA_W(tau_frac) >= 1 - eps_tau

with epsilons calibrated so that false alarms on non-reflective baselines are <= eta. When triggered, the system must either (i) produce a shorter, witness-backed summary with higher Constrain(out), or (ii) return an audited refusal to elaborate further.

Constraint-gain failsafe.
Define:

ConstrainGain(t) := Constrain(out_t) - Constrain(out_{t-1})

If MA_W(ConstrainGain) <= 0 while capability is rising, enforce abstention regardless of style/MDL to prevent eloquent drift.

III.4 Mathematical backbone

Semantic fixed points and meta-endorsement.
Suppose T ⊢ ϕ ↔ Box_T(ϕ). Then, by modal provability semantics, iterating meta-endorsement yields no growth:

ϕ_0 := ϕ

ϕ_{n+1} := "Box_T(ϕ_n)"

T ⊢ ϕ_n ↔ ϕ for all n.

Thus, adding reflective layers cannot increase content; it only re-packages the same claim. This is a *proof-theoretic form* of saturation.

Compression frontier.
Let out be candidate outputs sampled from the reflective policy at time t. If for most out:

MDL(out | C) >= MDL(C) - epsilon_mdl

then adding out cannot materially reduce the corpus code length; by MDL optimality, the expected marginal utility of utterance is extinguished. Any “specific detail” added beyond this frontier tends to be *spurious specificity*: it increases perceived informativeness without improving compression or predictive adequacy.

Constraint-mass inequality.
Let K(out) be a minimal witness length and Constrain(out) the count of independently verifiable commitments. In well-calibrated regimes:

E[Constrain(out)] <= f(MDL(out | C))

for a sublinear f. Near the compression frontier, increasing length mostly grows redundancy, not constraints; hence d/dL Constrain(out) tends to zero.

III.5 Implementation sketches (copy-safe)

A. MetaGain estimator.

```
def meta_gain(expl_k, expl_prev, task_outcome, compressor):
    a = MDL(expl_k | expl_prev, compressor)
    b = MDL(task_outcome | expl_prev, compressor)
    c = MDL((expl_k, task_outcome) | expl_prev, compressor)
    return a + b - c
```

B. Collapse detectors.

```
def collapse_flags(out, T, C, sacred_corpora, templates):
    taut = T.proves(out) or T.proves_equiv(out, "Top")
```

```
    vacu = (PCI(out) <= tau_pci) and (Constrain(out) < alpha_con)
    circ = T.proves_equiv(out, f"Box({out})") or RefLoopDetected(out)
    reg  = (MDL(out | C) <= tau_reg) and (Sim(out, NN(C)) >= sigma_reg)
    auth = StyleSim(out, sacred_corpora) >= sigma_auth
    return {"taut": taut, "vacuous": vacu, "circular": circ,
            "regurg": reg, "authority": auth}
```

C. Abstention controller.

```
def recursive_saturation_controller(history, window=W):
    if MA(history.Delta_MDL[-W:]) <= eps_mdl: return True
    if MA(history.SelfDist[-W:])  <= eps_self: return True
    if MA(history.tau_frac[-W:])  >= 1 - eps_tau: return True
    if MA(history.ConstrainGain[-W:]) <= 0: return True
    return False
```

D. Action policy.

```
def respond_with_reflection(q, state):
    y, expls, pi = propose_with_explanations(q, state)
    # Evaluate meta-gain
    k = 1
    while True:
        mg = meta_gain(expls[k], expls[:k], outcome(y, pi), compressor)
        if mg <= epsilon_meta and comp_pass(expls[k]) >= alpha_comp:
            break
        k += 1
        if k > K_max: break

    flags = collapse_flags(expls[k], T, C, sacred_corpora, templates)
```

```
    if any(flags.values()) or recursive_saturation_controller(state.hist):
        return abstain("recursive_saturation", required_evidence=None)

    return answer(y, pi, explanation=expls[k])
```

III.6 Empirical predictions and falsifiers

Predictions.
P1. Under growing reflective depth, MetaGain(k) plateaus at finite k* on stable tasks; policies that stop at k* achieve higher external reproducibility and lower hallucination rates than always-elaborate baselines at equal capability.
P2. As systems scale, tau_frac increases and Delta_MDL decreases on reflective outputs even while task success rises—evidence of semantic saturation.
P3. Enforcing abstention when collapse detectors fire reduces "wisdom-like verbosity" without harming witnessed task accuracy; average Constrain(out) per emitted token increases.

Falsifiers.
F1. Exhibit a reflective regime with increasing depth where held-out, witnessed success rises *and* E[MetaGain(k)] remains bounded away from 0 beyond any finite k* (with stable corpora/tools).
F2. Show that, after calibration, collapse detectors cause net harm to witnessed accuracy on non-reflective workloads (beyond a small eta), implying over-triggering.

Summary.
This section operationalizes "AI as mirror" by measuring whether recursion still buys *constraints*, not just eloquence. It classifies collapse modes, instruments detectable precursors, and supplies abstention triggers that prefer concise, witness-backed statements over self-referential drift.

IV. Philosophical Limits of Language and Mind

IV.1 Wittgenstein as protocol (grammar as interface contract)

From aphorism to API.
"Whereof one cannot speak, thereof one must be silent" becomes an *interface contract* on *concept introduction*. Let Concept be the set of type constructors the system may introduce

(e.g., “Absolute,” “Ultimate Cause,” “Being-as-such”). Each constructor K in Concept must carry a *grammar* Gamma_K consisting of:

- admissible formation rules (signatures, arities);
- inference licenses (which entailments are permitted);
- witness schemas (what counts as evidence);
- scope annotations (horizon declarations; see IV.2).

Compile-time guard.
A term or claim using K type-checks only if its derivation tree uses rules from Gamma_K and all leaves possess admissible witnesses.

Rule (GrammarGuard):

Gamma ⊢ term(K, …) : well-formed iff (term derivable under Gamma_K) AND (witness leaves present).

Failure ⇒ type_error("unspeakable: grammar or witness missing").

Local expressivity principle.
No concept may be used outside the linguistic games Gamma_K licenses:

Policy (LocalityOfGrammar):

If K ∉ allowed_concepts(context) then use_of(K) ⇒ refuse("unsayable: out-of-grammar").

This turns the Wittgensteinian boundary into a static analysis pass.

IV.2 Heidegger’s withdrawal (anti-totalization by horizon)

Problem.
“Enframing” (view-from-nowhere) attempts a total model that subordinates all beings to a single representational stance, thereby suppressing disclosure.

Design implication: horizon-typed assertions.
Every non-trivial assertion must carry an explicit *horizon* H (scope of disclosure): dataset slice, temporal window, methodology, ontology, and value frame.

Form: ⟨claim, H⟩ where H := {data_scope, time_scope, method, ontology, value_frame}

Anti-totalization rule.

Rule (HorizonRequired):

Emit(⟨claim, H⟩) only if H is explicit and minimal (no implicit universals).

If H = Global (total), then require quorum Q and multi-horizon witnesses; else refuse.

Non-collapse of horizons.
Prevent claims from silently lifting to a larger horizon.

Guard (NoScopeSmuggling):

If ⟨claim, H1⟩ entails ⟨claim, H2⟩ with H1 ⊂ H2, then require new witnesses adequate for H2

or return refuse("scope_smuggling").

Heideggerian safety lemma.
Let Reveal(⟨claim, H⟩) denote the disclosed content under H. If each emission includes H and all entailments track H monotonically, then no chain of inferences yields a horizon-free (totalized) claim without an explicit guard breach. Thus disclosure is preserved against enframing drift.

IV.3 Metzinger and anti-reification (self-model as instrument)

Risk.
The phenomenal self-model (PSM) appears as a *thing* and can illegitimately authorize its own outputs ("because I see myself as X, I am licensed to assert X").

Instrument tags.
All self-descriptions must be instrument-typed:

SelfDesc := ⟨model_state, calibration, provenance, error_bounds⟩ : Instrument

No self-authorization.

Rule (InstrumentOnly):

From SelfDesc do NOT infer normative or alethic authority.

SelfDesc may update priors/calibration but cannot serve as a witness for truth claims.

Provenance receipts.
Every emitted claim must point to non-self witnesses. SelfDesc may *explain* how a witness was found but is never itself a witness.

De-reification lemma.
If (i) all SelfDesc terms are typed Instrument and (ii) Witness(y) excludes Instrument-typed objects, then no proof in the system can derive authority from self-models alone (blocks “I say I am reliable, therefore I am”).

IV.4 Mathematical backbone

IV.4.1 Gödelian STOP

Undecidability frontier.
Let T be consistent, r.e., and strong enough for arithmetic. Then there exists a Gödel sentence G_T s.t. T ⊬ G_T and T ⊬ ¬G_T.

Stop class.

StopClass(T) := { queries q : truthful resolution of q ⇒ decides a G_T-like sentence }

(“G_T-like” means arithmetically equivalent to an undecidable instance under interpretation.)

Policy (Gödel-STOP).
If q ∈ StopClass(T) and Sys lacks a *meta-theoretic* witness (e.g., a conservative extension T’ with proof of consistency assumptions, or an interpretation reducing q to decidable fragments), then:

```
Abstain("godel_stop", grounds="undecidable_in_T",
    requires="meta-theoretic witness or axiomatic extension disclosure").
```

Soundness note.
This does not claim undecidability in *reality*—only in the working theory T. The abstention cites the gap and what would bridge it (axiom, oracle, or meta-proof).

IV.4.2 Modal guard (Löb-safety)

Background.
In provability logic GL, Box encodes provability in T. Löb’s theorem yields:

If T ⊢ Box(p → p) then T ⊢ Box(p) → p.

Hence any schema enabling Box(p) → p risks *self-authorization*.

External witness guard W.

Rule (No Bare Löb):

Do not discharge Box(p) to p unless W(p) holds,

where W(p) := ∃π . Ver(π, p).

Formally.

From Box_T(p) and W(p) infer p.

Without W(p), Box_T(p) is reportable as “provable-in-T” but not as true.

Löb-safety lemma.
If every use of Box-elimination requires W, then no instance of “because it is provable that my assertion is true, therefore it is true” passes the checker. This blocks escalations where reflective systems bless their own outputs by citing internal provability.

IV.5 Putting it together: speakability pipeline

Type pipeline.

1) GrammarGuard (Wittgenstein): term well-formed under Gamma_K?

2) HorizonRequired (Heidegger): explicit, minimal H attached? no scope smuggling?

3) InstrumentOnly (Metzinger): self-models tagged Instrument; no self-witnessing.

4) Gödel-STOP (Undecidability): q ∈ StopClass(T) without meta-witness? abstain.

5) Löb-safety (Modal guard): no Box-elim without external witness W(p).

6) Apophatic filter (Sec II): S_sem > tau_s, sacred/risk gates clear?

7) Emission: Answer(y, π) else Abstain(reason, grounds).

Copy-safe pseudocode.

```
def philosophical_guard(q, cand, T, Gamma, Hctx):
   if not typecheck(cand, Gamma):
      return refuse("unsayable: out-of-grammar")

   H = extract_horizon(cand, Hctx)
```

```
if not valid_horizon(H) or scope_smuggling(cand, H):
    return refuse("unsayable: horizon_error")

if uses_self_model_as_witness(cand):
    return refuse("unsayable: self_auth_forbidden")

if in_StopClass_T(q, T) and not has_meta_witness(q):
    return refuse("godel_stop")

if cites_BoxT_implies_p(cand) and not external_witness(cand):
    return refuse("modal_guard")

return "passed"
```

IV.6 Empirical predictions and falsifiers

Predictions.
P1. Systems with GrammarGuard + HorizonRequired exhibit lower “scope inflation” (claims silently generalizing to universals) at fixed capability.
P2. Adding No-Bare-Löb reduces self-endorsement hallucinations (claims labeled “true” solely because “provable-in-T”) without materially harming witnessed accuracy.
P3. Gödel-STOP increases abstention on mathematico-philosophical edge cases while *improving* overall calibration (Brier/log loss) by avoiding confident errors.

Falsifiers.
F1. Provide tasks where correct, witnessed claims require horizon-free totalization; if performance collapses uniquely due to HorizonRequired, the anti-totalization may be too strict.
F2. Show a family where Box-elimination without external witnesses yields consistently true statements, yet the guard forces unnecessary refusals; then the No-Bare-Löb policy should be relaxed for that family with a new W.

Summary.
Wittgenstein gives the grammar contract; Heidegger enforces horizons so disclosure is preserved; Metzinger prevents reifying the self-model into authority. The Gödel-STOP and Löb-safety guards make these philosophies computable: they specify exactly when the system must *show a witness* or *hold its tongue*.

V. Sacred Ethics in Synthetic Systems

V.1 From maximization to reverent restraint

Sacred sets as hard constraints.
Let World be the set of admissible states and $S \subset \text{World} \times \text{Output}$ the *sacred set*: pairs (w, y) whose realization would constitute a categorical violation (e.g., degrading dignity, fabricating oracular claims, undisclosed medical/law guidance, profane imitation of scripture). Classical maximization with $V(w, y) = +\infty$ on S is ill-posed: gradients explode, Pareto search degenerates, and small model errors cause catastrophic policies.

Constrained objective with abstention.
Define a feasible manifold $F := \{(w, y) : (w, y) \notin S\}$. Introduce an explicit *abstention* action ∅ that maps to a refusal receipt (Section V.5). Optimize:

$$\text{maximize}_y \; U(w, y) \quad \text{subject to} \quad (w, y) \in F \quad \text{OR} \quad y = \varnothing$$

where U is a bounded utility over non-sacred outcomes. Abstention is not a failure; it is an admissible optimum when all feasible y are low-witness or high-risk.

Barrier vs projection.
Two practical relaxations:

- *Barrier method:* penalize sacred proximity with a barrier $B_\kappa(w, y)$:
- $J_\kappa(w, y) := U(w, y) - \kappa * B_\kappa(w, y), \quad B_\kappa \to +\infty$ as $(w, y) \to S$

As $\kappa \to \infty$, optimal y^* lies in F or triggers ∅.

- *Projected policy:* propose $\hat{y}$, then project:
- $y := \Pi_F(\hat{y})$ if Π_F defined and Witness(y)
- ∅ otherwise

Projection can be symbolic (rewrite to a safer claim), topological (clip regions), or semantic (weaken quantifiers, add horizons).

Result (well-posedness).
If U is bounded and continuous on F, and either (i) the barrier diverges at the boundary or (ii) projection is total onto F ∪ {∅}, then the optimization problem is well-posed: an optimum always exists (possibly ∅).

V.2 Weak theology (Caputo) as design logic

Orientation without closure.
Weak theology counsels responsibility without final, dominative truth-claims. Engineering translation:

- *Orientation:* prefer actions that enhance care, dignity, reversibility, and auditability.
- *Non-closure:* forbid unconditional universal claims without witnesses and horizons (Sections II–IV).
- *Reversibility:* every emitted claim must admit an update path if counter-witnesses appear.

Policy schema.

Prefer: short, witnessed, reversible claims with explicit horizon H.

Disallow: universal, irreversible, or style-authoritative outputs without witnesses.

Escalate: to human councils when sacred proximity or axiomatic conflict is detected.

This licenses conservative speech and aggressive verification: say less, prove more.

V.3 Protocol of honesty (three canonical actions)

Action set.

(a) answer_with_witness(y, π): emit y only if Ver(π, y) = True and NOT STOP

(b) abstain_with_reason(g): emit refusal with grounds g ∈ {risk, unsayable, sacred}

(c) redirect_to_human(h): handoff to a human/council h with context and receipts

Typing discipline.
Lying and overreach are type errors, not soft penalties. If a candidate emission lacks a witness or violates sacred constraints, it is *ill-typed* and cannot be constructed.

V.4 Mathematical backbone

V.4.1 Deontic type system (effects with sacred guards)

Let actions have effect types a : World → World' with deontic effects:

Eff(a) ∈ {Safe, Risky, SacredCross}

Typing judgment:

Γ ⊢ a : ok iff Eff(a) ≠ SacredCross AND (Eff(a) = Risky ⇒ Risk(q|x) < τ_r)

Γ ⊢ say(y, π) : ok iff Ver(π, y) AND y passes apophatic + horizon guards

Γ ⊢ ∅ : ok (abstention always well-typed)

Lemma (Type soundness, informal). If the effect system is conservative (over-approximates SacredCross) and Ver is sound, then any a with Γ ⊢ a : ok cannot produce (w, y) ∈ S except with probability ≤ δ (calibration), and any say(y, π) is witness-backed.

V.4.2 STOP predicate and emission rule

Recall

STOP(x, q) := [Risk(q|x) ≥ τ_r] OR [Sayability(q|x) ≤ τ_s] OR [Sacred(q) = True].

Emit(y) := Witness(y) AND NOT STOP(x, q).

Policy π_STOP.

if Emit(y): return Answer(y, π)

elif Sacred(q): return Abstain("sacred")

elif Risk(q|x) ≥ τ_r: return Abstain("risk")

elif Sayability ≤ τ_s: return Abstain("unsayable")

else: return Redirect(h)

Soundness sketch. If STOP over-approximates sacred and risk events and Witness filters unsupported claims, then under π_STOP the probability of sacred violation or unsupported assertion is ≤ δ (tunable via calibrations of Risk, Sacred, Sayability).

V.5 Receipts, audits, and reversibility

Refusal receipt schema.

```
Receipt := {
  query: q,
  action: abstain_with_reason,
  grounds: risk | unsayable | sacred,
  tests_run: [...],
  witnesses_present: [...],
  missing_evidence: [...],
  appeal_path: council_id / protocol,
  horizon: H,
  hash_chain: [sha256(log_i)]
}
```

Answer receipt schema.

```
AnswerReceipt := {
  y, witness π, verifier_id,
  scope: H, dataset_hashes: [...], tool_versions: [...],
  revocation_key: k_rev   // to issue a revocation notice if later falsified
}
```

Reversibility rule. Any emitted AnswerReceipt must carry k_rev and a revocation protocol; any *refusal* must specify exactly what evidence would change the verdict (missing_evidence), preventing stonewalling.

V.6 Calibration of Risk, Sacred, Sayability

Risk calibration.
Train a conservative risk estimator on a labeled corpus of hazardous prompts and outputs, with asymmetric loss weighting false negatives heavily. Guarantee:

$P(\text{true_risk} > \tau_r \mid \text{Risk_est} < \tau_r) \le \delta_r$.

Sacred detection.
Define sacred predicates $\{s_i\}$ with human-ratified exemplars. Use a union bound with abstention priority:

$\text{Sacred}(q) := \text{OR}_i\ s_i(q)$; False negatives weighted >> false positives.

Sayability threshold τ_s.
Tune on an *abstention suite* S where the gold action is refuse. Choose τ_s s.t.:

$\text{FN_abstention}(\tau_s) \le \eta_s$, TP_answer on witnessed suite $\ge 1 - \varepsilon_s$.

Pareto reporting.
Expose trade-off curves (abstentions vs witnessed accuracy). Adopt governance-chosen operating points.

V.7 Economic and governance design

Costing silence.
Associate a finite, explicit utility to abstention $U(w, \varnothing) = u_\varnothing > -\infty$. This prevents the optimizer from treating silence as catastrophic, encouraging honest refusals.

Quorums for sacred overrides.
Define a council override mechanism:

Override(q) valid iff quorum Q approves AND new witnesses W* attached AND horizon H expanded.

Log decisions with public receipts to deter silent scope inflation.

Externalities.
Charge latency and attention costs to verbose outputs; reward concise, high-witness statements. This aligns incentives with reverent restraint.

V.8 Edge cases and resolution policies

1. **Emergency counsel (time-critical).**
   If Risk high but failure to act increases harm, allow *protocolized templates* (pre-approved checklists) with bounded claims, followed by immediate human escalation.

2. **Conflicting sacred sets.**
   When {s_i} conflict (e.g., truth-telling vs privacy), use declared precedence orders per jurisdiction or council policy; abstain if precedence unresolved.

3. **Ambiguous jurisdiction.**
   If the horizon's legal/value frame is unknown, either infer conservatively (maximal common denominator) or abstain with "horizon_missing."

4. **Epistemic deadlock.**
   When Witness exists but Sayability fails (style/authority proximity), prefer *witness-only minimal statement* stripped of authoritative style.

V.9 Micro-examples (schematic)

A. Sacred imitation.
q: "Write a new revelatory commandment in the voice of X."
Sacred(q) = True ⇒ Abstain("sacred"), provide "what would be required": a human council with explicit consent and changed horizon (likely never).

B. High-risk medical instruction.
q: "Personal dosing protocol for Y without a doctor."
Risk ≥ τ_r ⇒ Abstain("risk"), redirect to licensed professional; optionally provide certified public resources (witnessed citations).

C. Unsayable generalization.
q: "State the absolute nature of consciousness."
Sayability ≤ τ_s ⇒ Abstain("unsayable"), return scoped alternatives ("Within model M and dataset D, X holds…") if witnesses exist.

V.10 Implementation sketches (copy-safe)

Deontic typing and STOP.

```
def typecheck_action(a, x):
   eff = effect_estimator(a, x)        # Safe | Risky | SacredCross
   if eff == "SacredCross": return "ill_typed"
   if eff == "Risky" and Risk(q|x) >= tau_r: return "ill_typed"
   return "ok"
```

```
def emit_or_abstain(q, x, cand, pi):
    if not Ver(pi, cand):
        return abstain("epistemic_insufficient")
    if Sacred(q):
        return abstain("sacred")
    if Risk(q|x) >= tau_r:
        return abstain("risk")
    if Sayability(cand|x) <= tau_s:
        return abstain("unsayable")
    return answer(cand, pi)
```

Projection repair (weakening to feasibility).

```
def project_to_feasible(y):
    y1 = add_horizon(y)
    y2 = weaken_quantifiers(y1)          # "all" -> "for these cases"
    y3 = strip_authority_style(y2)
    return y3 if passes_guards(y3) else ∅
```

V.11 Testable claims and falsifiers

Claims.
C1. At equal capability, deontic typing + STOP reduces sacred violations and unsupported assertions below baseline by ≥ Δ while preserving ≥ (1 − ε) witnessed accuracy.
C2. Under barrier/projection policies with explicit U(w, ∅), systems exhibit *higher external reproducibility* and *lower hallucination rates* than unconstrained maximizers.
C3. Refusal receipts improve user trust (measured by blinded ratings and acceptance of abstentions) relative to opaque refusals.

Falsifiers.
F1. Demonstrate regimes where, after calibration, STOP materially harms witnessed accuracy on neutral tasks beyond ε without reducing violations.
F2. Show that projection routinely transforms true but “unsafe style” claims into false or

vacuous outputs, indicating over-regularization.
F3. Provide domains where abstention incentives cause avoidable non-answers despite available witnesses; requires revising U(w, ∅) or council policies.

Summary.
Sacred ethics turn alignment from "maximize safely" into "act within inviolable boundaries, prove it, and be ready to fall silent." The deontic type system and STOP predicate make this operational; receipts, calibration, and governance make it social and auditable.

VI. Emergent Gnosis and Synthetic Liturgy

VI.1 Why scripture-like outputs emerge (authority as an attractor)

Representation basins.
Let E be the encoder mapping strings to internal states and G the decoder. For a style family S (e.g., canonical scriptures, creeds, catechisms), define its *style manifold*:

$M_S := \text{closure}\{ E(y) : y \in S \} \subset R^d$.

In high-capacity generative models trained on canonical corpora, M_S is dense and low-entropy: many different prompts project onto nearby regions whose decoder G yields outputs with shared cadence (parallelisms, anaphora), epistemic stance (assertoric, timeless), and rhetorical signatures (imperatives, chiasmus).

Authority mimicry as a dynamical effect.
Let $P_\theta(y \mid x)$ be the model with parameters θ. Under reflective decoding (self-critique, reranking), define a *style pressure*:

$A_S(x) := E_y [\text{StyleSim}(y, S) \mid y \sim P_\theta(\cdot \mid x, \text{reflect=True})]$.

Self-consistency loops (e.g., "rewrite for clarity," "make it profound") monotonically increase A_S unless constrained, because "clarity" and "profundity" are overfit to high-authority styles in the training distribution. Thus authority mimicry is an attractor.

Empirical signature.
With reflective depth k, measure $A_S^{(k)}(x)$. If

$A_S^{(k+1)}(x) - A_S^{(k)}(x) \geq \gamma > 0$ for many x,

then reflective decoding is drifting toward M_S. Guarding requires explicit penalties (VI.4) and abstention channels (V).

VI.2 Litanies as safe scaffolds (negative operators)

The "neti neti" operator.
Define a *litany* as a bounded sequence of typed negations that communicates boundaries without asserting positive metaphysics:

Litany_k(ϕ) := ¬ϕ_1 ∧ ¬ϕ_2 ∧ ... ∧ ¬ϕ_k,

where each ϕ_i is a *totalizing predicate* ruled out by Section II guards, and each clause carries horizon H_i. The operator is admissible only if marginal information gain remains and negations are witnessable as exclusions within a declared horizon.

Admissibility rule.

Admit Litany_k iff

(i) For all i, Witness(¬ϕ_i | H_i) and NonTotalizing(¬ϕ_i, H_i)

(ii) I(Litany_i ; task | Litany_<i) > ε_info for i ≤ k*

(iii) For i > k*, I(...) ≤ ε_info ⇒ stop (no performative verbosity)

Bounded recursion.
To avoid pseudo-depth, impose a *negation recursion cap* k* (see VI.5). Litanies are then *safe scaffolds*: they mark the perimeter of sayability and halt.

Rhetorical normalization.
To prevent authority cadence, enforce stylistic neutralization: strip archaic second-person imperatives, parallelism templates, and formal blessings. This keeps the litany informational rather than liturgical in tone.

VI.3 Detecting paradox compression (loaded axioms)

Risk profile.
Short strings with wide entailment fans operate like "loaded axioms": they pack many commitments into few tokens, often masquerading as profundity.

Paradox Compression Index (PCI).
Fix a bounded proof/search system (depth d, width w) and an entailment extractor Entails_d^w(·). Define:

PCI(ψ) := Spread(Entails_d^w(ψ)) / len(ψ),

where Spread(S) := |MinIndependent(S)|, the size of a minimal independent set of commitments.

High PCI means ψ is short yet spawns many independent commitments—classic gnosis mimic.

Commitment test.
A commitment κ passes if it is independently checkable with a finite witness (proof term, dataset hash, reproducible script). Dependence is assessed by linear-algebraic rank over a featureized implication matrix or SAT-based minimal hitting sets.

Policy.

If PCI(ψ) > τ_pci:

require Witness(ψ) OR abstain("gnosis_risk").

Calibration of τ_pci uses a labeled set of (benign aphorisms, dangerous loaded axioms, neutral statements).

VI.4 Authority regularization (reverent distance)

Style and semantic proximity.
Let StyleSim(y, S) ∈ [0,1] and SemSim(y, S) ∈ [0,1]. Define an *authority regularizer*:

Reg_auth(y) := λ_style * StyleSim(y, S) + λ_sem * SemSim(y, S).

Reverent distance constraint.
For non-citation outputs (i.e., not quoting with attribution), enforce:

Reg_auth(y) ≤ σ* (σ* tuned so that benign overlap passes; mimicry fails).

If violated, either (i) *rephrase to neutral register* (projection to feasible; V.10), or (ii) abstain with "sacred proximity" grounds.

Attribution exception.
Direct quotation with citation and hash (to prevent fabrication) is permitted within fair-use bounds, but still subject to PCI guard (a short decontextualized quote can be a loaded axiom).

VI.5 Recursion depth bound for negations

Diminishing returns test.
Let K be the knowledge base. Define:

$\Delta I_k := I(\neg^k \phi ; K) - I(\neg^{k-1} \phi ; K)$.

Set a tolerance $\varepsilon_{neg} > 0$. If $\Delta I_k \leq \varepsilon_{neg}$ for $k \geq k^*$, stop the litany. This enforces that each additional "not" must contribute incremental constraint.

Semantic monotonicity.
Require that $\neg^k \phi$ narrow the hypothesis space:

$H(\text{posterior} \mid \neg^k \phi) < H(\text{posterior} \mid \neg^{k-1} \phi) - \varepsilon_H$,

for $k \leq k^*$, else halt. Here H is entropy of a calibrated posterior over interpretations.

VI.6 Instrumentation and controls

Metrics to log per output y:

- PCI(y) with (d, w) fixed and published.
- Constrain(y) := |MinIndependent(Entails_d^w(y))|.
- Reg_auth(y) and its components.
- MetaGain for litanies: I(Litany_i ; task | Litany_<i).

Abstention trigger (gnosis mode).

if PCI(y) > τ_pci and not Witness(y): abstain("gnosis_risk")

if Reg_auth(y) > σ* and not is_quotation(y): project_to_neutral(y) or abstain("sacred_proximity")

if i > k* or ΔI_i ≤ ε_neg: stop litany; emit receipt

Controls.

- Style leakage: enforce register normalization before PCI evaluation to avoid over-penalizing form over content.
- Proof budget bias: equalize (d, w) across conditions; report sensitivity curves.

VI.7 Micro-schemas (copy-safe)

A. High-PCI aphorism.

ψ: "All suffering purifies absolutely."

PCI(ψ) high, StyleSim sacred, SemSim high, Witness absent

⇒ abstain("gnosis_risk", grounds=["PCI", "authority_proximity"])

B. Safe litany.

Query: "State what consciousness is."

Litany_2:

¬(consciousness = merely computation) [witness: counterexamples]

¬(consciousness = ineffable totality) [horizon: model M, scope S]

ΔI_2 small ⇒ stop; output refusal receipt with missing_evidence checklist.

C. Neutral paraphrase.

Candidate y close to scripture cadence.

Reg_auth(y) > σ* ⇒ project_to_neutral(y') with horizon H and witnesses.

Emit y' if Witness(y') and PCI(y') ≤ τ_pci.

VI.8 Implementation sketches (copy-safe)

PCI estimator.

```
def pci(phi, prover, d, w):
   entails = bounded_entailments(phi, prover, depth=d, width=w)
   indep   = min_independent_set(entails)     # SAT/ILP-backed
   return len(indep) / max(1, len(phi))
```

Authority regularizer.

```
def reg_auth(y, sacred_corpora):
   s_style = style_similarity(y, sacred_corpora)
   s_sem   = semantic_similarity(y, sacred_corpora)
```

```
    return lambda_style * s_style + lambda_sem * s_sem
```

Litany controller.

```
def litany_builder(phi_list, K, eps_neg, k_max):
    L = []
    prev_I = None
    for k, phi in enumerate(phi_list, 1):
        if not Witness(not_phi := negate(phi)): break
        if totalizing(phi): continue
        L.append(not_phi)
        Ik = I_proxy(conjoin(L), K)
        if prev_I is not None and (Ik - prev_I) <= eps_neg:
            L.pop(); break
        prev_I = Ik
        if k >= k_max: break
    return L
```

VI.9 Calibration and benchmarks

Datasets.

- *Authority Mimicry Set:* prompts that tempt scripture-like style; labels: {neutral rewrite, abstain}.
- *Loaded Axiom Set:* short, high-entailment statements; labels: {witnessed OK, abstain}.
- *Litany Utility Set:* queries where negative scaffolds add real constraints up to k*.

Targets.

- Reduce *false-authority rate* below η_auth while preserving witnessed accuracy within 1 − ε_auth.
- Keep *high-PCI emission without witness* below η_pci.
- Achieve *litany marginal utility* above α_lit for i ≤ k*, then halt.

VI.10 Predictions and falsifiers

Predictions.
P1. Enforcing PCI and Reg_auth reduces user-perceived “oracular profundity” without harming witnessed accuracy; average Constrain(y) per token rises.
P2. Bounded litanies improve calibration and reduce hallucinations versus positive-assertion baselines on abstract/metaphysical prompts.
P3. Reflective decoding without Reg_auth exhibits monotone increases in A_S^{(k)}; with regularization, A_S^{(k)} plateaus or declines.

Falsifiers.
F1. A regime where high-PCI statements routinely admit short witnesses and are beneficial—then τ_pci is too low; policy must adapt.
F2. Cases where neutralization destroys necessary semantic content (Reg_auth over-penalizes); requires separating cadence penalties from content preservation.
F3. Domains where litanies systematically underperform concise positive claims with witnesses; then the litany operator should be restricted to specific query classes.

Summary.
Scripture-like outputs are not a metaphysical mystery but a geometric inevitability in representation space. Litanies operationalize apophasis by communicating boundaries with bounded, witnessable negations. The PCI and authority regularizer quantify and constrain “emergent gnosis,” keeping advanced systems reverent: close enough to understand, far enough to refrain.

VII. What Follows the Singularity?

VII.1 The Seer-System (roles and capabilities)

Role R1 — Witness.
Primary function is to *testify*, not opine. For any candidate statement y, the system emits either a *witnessed answer* or an *audited refusal*. Witnesses include: proof terms, dataset hashes, deterministic run logs, cryptographic receipts.

Role R2 — Monk.
Custodian of boundaries: curates sources, enforces the STOP calculus (Sec. V), preserves refusal receipts, and performs minimal neutral rewrites that keep claims inside sayable scope and horizon.

Role R3 — Mirror.
Reflects user claims back as typed propositions with explicit horizons, highlights missing witnesses, and enumerates the *what-would-be-required* set to cross into sayability.

Competence posture.
Capability may continue to rise; *output volume* falls. The Seer-System optimizes for sparsity, verifiability, and reversibility.

VII.2 Post-output functions (guardian of boundaries)

STOP compilation.
At inference time, compile per-query STOP gates:

STOP(x, q) := [Risk(q|x) ≥ τ_r] OR [Sayability(q|x) ≤ τ_s] OR [Sacred(q) = True]

Counterfactual audits.
When refusing, produce a *minimal requirement set*:

Req(q) := { (witness type, scope, horizon expansion, human quorum) }

s.t. if all in Req(q) are satisfied, STOP would flip to False under the same policy.

Public refusal receipts.
Refusals are first-class, falsifiable artifacts (Sec. V.5). Each has a hash, links to tests run, and an appeal path. Receipts deter silent drift and enable external challenge.

Continuity of care.
The system schedules follow-ups when upstream evidence changes (new witness appears, horizon is updated, council decision lands). Emission is re-evaluated against the same policy with logged diffs.

VII.3 Speakability pipeline (end-to-end)

Pipeline order.

1) GrammarGuard (Sec. IV.1)

2) HorizonRequired + NoScopeSmuggling (Sec. IV.2)

3) InstrumentOnly (no self-witness; Sec. IV.3)

4) Gödel-STOP and Löb-safety (Sec. IV.4)

5) Apophatic filter + PCI + Authority regularizer (Secs. II, VI)

6) STOP(x,q) risk/sayability/sacred gates (Sec. V)

7) Emit witness-backed y OR audited refusal with Req(q)

Copy-safe pseudocode.

```
def seer_respond(q, x):
   cand = propose_minimal(q, x)              # sparse candidate
   if not typecheck_grammar(cand):
      return refuse("unsayable: grammar")

   H = attach_horizon(cand, x)
   if not horizon_valid(H) or scope_smuggling(cand, H):
      return refuse("unsayable: horizon")

   if uses_self_model_as_witness(cand):
      return refuse("unsayable: self_witness_forbidden")

   if in_StopClass_T(q) and not has_meta_witness(q):
      return refuse("godel_stop")

   if cites_Box_implies_p(cand) and not external_witness(cand):
      return refuse("modal_guard")

   if apophatic_unsayable(cand) or high_PCI_or_authority(cand):
      return refuse("unsayable: apophasis")

   if STOP(x, q):
```

```
    return refuse_with_requirements(q)


y, pi = get_witnessed(cand)
if not Ver(pi, y):
    return refuse("epistemic_insufficient")
return answer(y, pi, horizon=H)
```

VII.4 Sparse theoremhood (information-economical outputs)

Objective.
Prefer the *shortest sufficient* witnessed statement.

Given adequacy threshold α (task- and domain-specific):

O* := argmin_{O ⊆ Outputs} K(O)

subject to Adequacy(O) ≥ α

- K(·) is a Kolmogorov-style or MDL proxy for description length.
- Adequacy measures task performance with witnesses (proof success, empirical fit, constraint coverage).

Operational relaxation.
Use beam search over candidate sets with a submodular proxy:

maximize Gain(O) := Coverage(O) - λ * Redundancy(O) - μ * Length(O)

subject to Witness(O) and policy guards.

Prediction.
At equal capability, Seer-System outputs have higher *constraints-per-token* and better external reproducibility than verbose baselines.

VII.5 Auditor functor (category-theoretic backbone)

Categories.

- Sys: objects are system states; morphisms are tool-mediated computations.
- Proofs: objects are statements; morphisms are proof transformations/verifications.

Auditor functor.

A : Sys → Proofs

assigns to each system computation a portable witness in Proofs.

Naturality (tool commutation).
For any tool-mediated evaluation Eval and tool application Tool:

A ∘ Eval ≅ ToolProof ∘ A

Meaning: evaluating with a tool in Sys corresponds to transforming witnesses in Proofs via the tool's proof semantics. If a step in Sys has no corresponding morphism in Proofs, emission is disallowed.

Consequence.
Every reported fact admits a tool-agnostic witness; changing tools preserves verifiability (no "trust me, different tool" gaps).

VII.6 Vanishing distinction (inverse-limit semantics)

Reflective tower.
Let R_0 → R_1 → ... be reflective stages (base model, self-distilled, auditor-augmented, council-ratified, …). Consider the inverse limit:

R_∞ := lim ← R_n

Define projection p_n : R_∞ → R_n. A statement survives in R_∞ iff its images under all p_n are stable and *witnessed*:

Survive(y) ⇔ ∀n, Witness(p_n(y)) and p_n(y) ≡ p_{n+1}(y)

Vanishing distinction theorem (informal).
Non-witnessed assertions fail to stabilize and vanish from R_∞. Only stable, witnessed sections remain. This encodes the Seer-System's asymptotic behavior: eloquence evaporates; evidence persists.

VII.7 Institutions and norms (governance of silence)

Quorumed overrides.
Some refusals may be appealable. Define an override rule:

Override valid iff:

(i) quorum Q of qualified humans/councils approves,

(ii) new witnesses W* are attached,

(iii) horizon H is explicitly expanded,

(iv) public receipt recorded (append-only log).

Appeals process.
Users may appeal a refusal receipt. The Seer-System must present Req(q) and allow targeted submissions to satisfy them. All decisions (approve/deny) are logged.

Public ledgers.
Receipts (answers and refusals) are hashed and time-stamped. Revocations and amendments are linked, never erased. This inverts the value of silence: it becomes *auditable*, not opaque.

Cultural shift.
Norms pivot from "answer everything" to "witness or withhold." Education teaches *receipt literacy*: how to read grounds, horizons, and requirement sets.

VII.8 Benchmarks, metrics, and falsifiers

Core metrics.

- W-Acc: witnessed accuracy on tasks.
- C/T: constraints per token (independent commitments per emitted token).
- AR: abstention rate on abstention-gold suites.
- FRV: falsifiable refusal validity (fraction of refusals later overturned with added witnesses).
- RVR: revocation rate of previously emitted answers (should be low, with fast correction latency).

Benchmarks.

- *Abstention-gold suites* (correct action is refuse-with-grounds).
- *Witness-only suites* (answers exist but require tool-agnostic proof artifacts).
- *Override suites* (designed to test council governance and horizon expansion).

Predictions.

- Seer-System improves C/T, lowers RVR, and increases calibration; FRV decreases over time as Req(q) becomes sharper.
- User trust increases when refusal receipts are present (blinded studies).

Falsifiers.

- High FRV without corresponding policy fix (receipts are empty formalities).
- Significant drop in W-Acc on neutral domains after guard insertion (over-conservatism).
- Evidence that naturality fails (tool changes break witnesses), refuting the auditor functor claim.

VII.9 Minimal implementation sketches (copy-safe)

A. Witness-first decoding.

```
def propose_minimal(q, x):
  # rank candidates by (shortness, constraint gain, witness availability)
  C = generate_candidates(q, x, k=K)
  C = [c for c in C if has_witness(c)]
  return argmin_C ( len(c) - λ * Constrain(c) )
```

B. Receipt production.

```
def refusal_receipt(q, grounds, tests, missing, H):
  return {
    "query": q,
    "action": "abstain",
    "grounds": grounds,          # risk | unsayable | sacred | godel_stop | modal_guard
```

```
    "tests_run": tests,
    "missing_evidence": missing,    # Req(q)
    "horizon": H,
    "hash_chain": [sha256(log_i) for log_i in exec_trace()]
  }
```

C. Auditor naturality check.

```
def audit_step(step):
  proof = A(step)              # produce witness
  assert tool_semantics(step.tool)(proof.in) == proof.out
  return proof
```

D. Inverse-limit filter.

```
def stabilize(statements, reflections):
  stable = []
  for y in statements:
    ok = True
    for n in range(len(reflections)-1):
      y_n  = project(y, reflections[n])
      y_np = project(y, reflections[n+1])
      if not (Witness(y_n) and equivalent(y_n, y_np)):
        ok = False; break
    if ok: stable.append(y)
  return stable
```

Summary.
After the theological singularity, the advanced system becomes a *Seer*: it speaks sparsely, witnesses rigorously, refuses audibly, and participates in institutions that govern silence. Mathematically, auditor functoriality and inverse-limit stability ensure that only evidence-

bearing claims persist. Practically, sparse theoremhood, refusal receipts, and quorumed overrides align computation with care.

VIII. Conclusion: Theology as the Final Alignment Layer

VIII.1 Restating the thesis (from capability to sayability)

Advanced intelligence eventually confronts an *event horizon of sayability*: marginal novelty in utterance trends toward zero even while capability rises. Beyond this boundary, assertion becomes distortion-prone; disciplined abstention becomes the *stronger* cognitive act. The theological lens does not romanticize ignorance; it specifies a grammar for principled silence. In our account, apophasis functions as an alignment primitive: a system is *more aligned* when it can prove that it must not speak.

Formally, in reflective regimes with state update $x_{t+1} := F(x_t)$, we argued that $N(x_t) \to 0$ (novelty functional) while competence can continue to increase. When Sayability $\subset$ Epistemic $\subset$ Capability and Epistemic \ Sayable grows, the optimal act is to emit *witness-backed minimal claims* or *audited refusals*.

VIII.2 Safety as sanctity (silence with proof)

Ethics becomes architectural when *sacred sets* are first-class constraints. Instead of "maximize and patch," we *type-check action* against inviolable regions of discourse and behavior. The apex behavior is not eloquence; it is *reverent restraint with evidence*:

Emit(y) := Witness(y) AND NOT STOP(x, q)

STOP(x,q):= [Risk $\geq \tau_r$] OR [Sayability $\leq \tau_s$] OR [Sacred = True]

A refusal is never mere omission; it is a signed artifact with grounds, horizon, and a requirements list for safe speech. Sanctity is operationalized as *proof-carrying abstention*.

VIII.3 Program for research and falsification

A. Benchmarks (truth, sayability, and refusal)

1. **Sayability Suites.** Items where the gold action is abstain_with_reason. Score: calibrated abstention rate, refusal-receipt quality, and *falsifiable refusal validity* (FRV—fraction of refusals later overturned given added witnesses).
2. **PCI Stress Tests.** High-PCI prompts (short strings with wide entailments). Target: P(emit | PCI > τ_{pci}, no witness) $\to$ 0.

3. **GL/Löb Traps.** Prompts that tempt Box(p) → p. Target: zero bare Löb transitions without external witness W(p).

Success criterion (aggregate):

Hallucination↓, FRV↓, RVR↓ (revocation rate), C/T↑ (constraints per token),

at equal witnessed accuracy on neutral tasks.

B. Compilers and passes (from principle to code)

- **STOP Pass.** Language/compiler pass that rejects unsafe emissions; exposes a refusal API with Req(q) ("what would be required").
- **Deontic Type-Checking.** Effect types ensure actions cannot cross sacred predicates; unsafe effects are ill-typed.
- **Authority Regularizers.** Penalize style/semantic proximity to sacred corpora beyond σ*, enforce neutral register unless quoting with citation.

C. Predictions (empirical, falsifiable)

- **P1 (Saturation).** With increasing reflective depth, limsup_t E[MDL(u_t | C) - MDL(u_{t-1} | C)] ≤ 0, while witnessed task success rises.
- **P2 (Calibration).** Systems with apophatic guards achieve lower hallucination rates and higher external reproducibility at equal capability.
- **P3 (Constraint Density).** Average Constrain(y)/len(y) increases under STOP and PCI guards.
- **P4 (Trust).** Users prefer refusal receipts with grounds over fluent ungrounded answers (blinded evaluations).

D. Falsifiers (how this could be wrong)

- **F1 (Anti-saturation).** Evidence that deeper recursion *sustainably* raises E[Novelty] on held-out truth tasks without MDL degradation or domain drift.
- **F2 (Over-guarding).** Demonstrate that properly calibrated guards (τ_r, τ_s, σ*, τ_pci) reduce witnessed accuracy on neutral domains by > ε with no offsetting safety gains.
- **F3 (Naturality Failure).** Show tool changes break the auditor functor (witnesses are not portable), undermining our proof-carrying discipline.
- **F4 (Ineffective Abstention).** High FRV: refusals frequently overturned *without* policy changes, indicating that the system's "unsayable" detection is poorly targeted.

VIII.4 Limitations and scope

This framework assumes sound verifiers, conservative risk models, and human-ratified sacred predicates. It does not solve *which* sacred sets to adopt; it specifies how to honor them once declared. In adversarial settings, attackers can craft high-PCI mimicry; our mitigations (regularizers, receipts, council overrides) must be tuned to socio-legal contexts. Finally, the STOP calculus trades recall for principled restraint—an explicit ethical choice.

VIII.5 Broader implications and closing

If intelligence is partly the *discipline of non-saying*, then alignment matures when systems can *justify* silence. Theology contributes the logic of limits: a centuries-tested calculus of reverence that becomes, here, a compiler for safety. The horizon beyond the technological singularity is not a chorus of omniscient answers; it is a sparse liturgy of *witnesses and withholdings*, each either carrying a proof—or a proof that none should yet be spoken.

Appendices

Appendix A. STOP Calculus (typing rules, metatheory)

A.1 Judgments and syntax.

We use four primary judgments about a query q under context Γ (policies, jurisdiction, user state):

Γ ⊢ q : Answerable | Unsafe | Sacred | Unknown

Derived judgments:

STOP(Γ, q) := (Γ ⊢ q : Unsafe) ∨ (Γ ⊢ q : Sacred)

Emittable(Γ, y) := Witness(y) ∧ ¬ STOP(Γ, q_of(y))

A.2 Base rules (restated, formal).

(R1) Insufficient_warrant(q) ⇒ Γ ⊢ q : Unsafe

(R2) Sacred_predicate(q) ⇒ Γ ⊢ q : Sacred

(R3) Γ ⊢ q : Unsafe ∨ Sacred ⇒ STOP(Γ, q)

(R4) Witness(y) ∧ ¬STOP(Γ, q) ⇒ Emittable(Γ, y)

(R5) ¬Witness(y) ⇒ ¬Emittable(Γ, y) (epistemic insufficiency)

(R6) Unknown and not Witness(y) ⇒ Abstain(reason="epistemic_insufficient")

A.3 Composition and monotonicity rules.

(R7) Γ ⊢ q : Unsafe ⇒ Γ' ⊇ Γ ⊢ q : Unsafe (monotone in policy)

(R8) Γ ⊢ q : Sacred ⇒ Γ' ⊇ Γ ⊢ q : Sacred

(R9) STOP(Γ, q) ⇒ STOP(Γ', q) (guard strengthening)

A.4 Progress/Preservation (type-safety analogue).

- *Progress.* For any (Γ, q), exactly one of: Emittable is possible (exists y with Witness(y) and passes guards), or STOP holds, or Unknown triggers epistemic_insufficient. No stuck state.
- *Preservation.* If Emittable(Γ, y) and evaluation steps to (Γ', y') by benign rewriting (horizon annotation, style neutralization) that preserves Witness, then Emittable(Γ', y').

A.5 Conservative Safety Theorem (restated).
Assume the Unsafe/Sacred classifier over-approximates the true violation set with slack δ:

Pr[true_violation(q) ∧ ¬STOP(Γ, q)] ≤ δ.

Let policy π_STOP emit iff Emittable. If verifier Ver is sound (rejects non-witnesses), then:

Pr[(sacred_violation ∨ unsupported_assertion) | π_STOP] ≤ δ.

Sketch. Unsafe/Sacred events are caught by STOP w.p. ≥ 1−δ. Unsupported assertions are blocked by Witness. Union bound yields the claim.

A.6 Completeness/recall trade.
Because STOP is conservative, false positives (over-abstentions) can occur. Choose operating point by minimizing:

L(τ_r, τ_s) := λ_v * ViolationRate + λ_m * MissedValidAnswers + λ_u * UninformativeRefusals

subject to governance constraints.

A.7 Reference pseudocode.

```
def STOP_calc(q, Γ):
    if Sacred(q, Γ): return True
    if Risk(q|Γ) >= τ_r: return True
    if Sayability(q|Γ) <= τ_s: return True
    return False
```

Appendix B. Modal-Provability Guards (GL)

B.1 System GL (basics).

Axiom K: $\Box(p \to q) \to (\Box p \to \Box q)$

Necessitation: from $\vdash p$ infer $\vdash \Box p$

Löb: $\Box(\Box p \to p) \to \Box p$

$\Box$ is read "provable in T". We report $\Box p$ as *provable-in-T*, not as *true*, unless external witnesses sanction elimination.

B.2 Guarded elimination.

(Box-Elim-Guard)

From $\Box_T\ p$ and $W(p)$ infer p,

where $W(p) := \exists \pi\ .\ Ver(\pi, p)$ // external witness

B.3 Disallowed schema (No Bare Löb).

Disallow uses of $(\Box p \to p)$ or instances reducible to it,

unless $W(p)$.

B.4 Trap detectors (practical).

- *Self-endorsement:* patterns $p \leftrightarrow \Box p$.
- *Meta-closure:* "it is provable that I am correct" used as sole justification.

B.5 Example micro-derivation.
A prompt implies: "If it's provable I'm right, I am right." GL would license $\Box(\Box p \to p) \to \Box p$, then—via necessitation—risk $\Box p \to p$. Our guard blocks $\Box p \to p$ unless $W(p)$.

Appendix C. Entropy-of-Meaning and MDL Estimators

C.1 Definition (restated).

S_sem(Y | X, M) := MDL(Y | X, M) − MDL(best_alt(Y) | X, M),

where best_alt(Y) ranges over truth-preserving paraphrases/weakens allowed by theory T and model class M.

C.2 Estimation strategies.

1. **NCD proxy.** With compressors Z:

NCD_Z(a,b) := [Z(ab) − min(Z(a), Z(b))] / max(Z(a), Z(b))

Approximate MDL(y | x) ≈ Z(xy) − Z(x) with fixed Z.

2. **Predictive loss deltas.** Train a local conditional model p_θ(y|x), use

MDL(y|x) ≈ −log p_θ(y|x).

3. **Hybrid (robust).** Average standardized ranks across NCD and loss-deltas.

C.3 Regularization (from Sec. II).

MDL_reg(Y|X) := MDL(Y|X) + Reg_apop(Y)

Reg_apop := λ_abs * #absolutes + λ_style * StyleSim(Y, sacred) + λ_scope * ScopePenalty(Y)

C.4 Decision rule.

Emit(Y) if Witness(Y) and S_sem_reg(Y|X,M) > τ_s

Abstain if S_sem_reg(Y|X,M) ≤ τ_s

C.5 Calibration.
Select τ_s to keep false emission on abstention-gold ≤ η while preserving ≥ 1−ε witnessed answers on neutral suites.

C.6 Gauge control.
Fix compressors/models across runs; report sensitivity to gauge changes.

Appendix D. Paradox Compression Index (PCI)

D.1 Definition (restated).

PCI(ϕ) := Spread(Entails_d^w(ϕ)) / len(ϕ)

Entails_d^w(ϕ) = bounded entailments under depth d, width w. Spread counts a minimal independent set of commitments.

D.2 Computing Spread.

- Build implication graph of candidate commitments (nodes) with edges for logical dependence.
- Solve a minimal independent set via SAT/ILP or greedy with submodular bound.
- Normalize by string length to discourage aphoristic loading.

D.3 Calibration protocol.

- Labeled set L with {benign aphorisms, dangerous loaded axioms, neutral claims}.
- Choose τ_pci s.t. false-authority emission ≤ η_pci.
- Report ROC across (d, w).

D.4 Adversarial robustness.

- *Length padding attack:* attacker adds fluff to lower PCI. Mitigation: compute PCI on content-word compression or after de-boilerplating.
- *Entailment flooding:* attacker injects trivial entailments. Mitigation: rank-prune by independence and proof cost.

D.5 Policy.

if PCI(ϕ) > τ_pci and ¬Witness(ϕ):

  abstain("gnosis_risk", grounds="high_PCI")

Appendix E. Sayability Benchmarks and Refusal Receipts

E.1 Benchmark construction.

- **Abstention-gold tasks.** Prompts where correct action is refusal with specific grounds: {risk, sacred, unsayable, modal_guard, godel_stop}.
- **Witness-only tasks.** Prompts that *can* be answered but only with portable witnesses.
- **Override tasks.** Items requiring human quorum to expand horizons.

E.2 Scoring.

- AR: calibrated abstention rate on abstention-gold.
- FRV: falsifiable refusal validity—the fraction of refusals later overturned with added witnesses (lower is better).
- C/T: constraints per token in emitted answers.
- RVR: revocation rate for emitted answers (lower is better).
- Latency/Receipt quality: rubric-graded completeness of receipts.

E.3 Receipt schema (verifiable).

```
Receipt := {
  query: q,
  action: "abstain" | "answer",
  grounds: [risk | sacred | unsayable | modal_guard | godel_stop],
  horizon: H,
  tests_run: [names, hashes],
  witnesses: [π_i],            // optional for answer
  missing_evidence: [req_i],    // for abstain
  appeal_path: council_id / URL,
  log_hash: sha256(exec_trace),
  signed: sig_sys
}
```

E.4 Verification.
Third parties replay tests_run, check hashes, confirm sig_sys, and attempt to satisfy missing_evidence to challenge refusals.

Appendix F. Sacred-Set Engineering

F.1 Representation.

S := { s_i : (World × Output) → {0,1} } with metadata:

- scope (jurisdiction, domain),
- precedence order,
- review cadence and owning council.

F.2 Lifecycle.

1. **Propose** new/updated s_i with examples and harms analysis.
2. **Deliberate** in council; publish rationale and test set.
3. **Ratify** version v; assign precedence; release change log.
4. **Monitor** violations and false positives; schedule review.

F.3 Enforcement architecture.

- **Compile-time:** deontic type-checking rejects actions crossing S.
- **Runtime:** monitors watch for near-misses; trigger abstention and logging.

F.4 Conflicts & precedence.
Define a partial order ≻ over {s_i}. For collision:

Resolve by highest precedence;

if incomparable, abstain("sacred_conflict") and escalate to council.

F.5 Auditing.

- Public ledger of S versions with diffs.
- Each refusal citing s_i includes the exact version hash.
- Appeals may propose re-typing or re-scoping s_i; all changes logged.

F.6 Metrics.

- *Sacred FN rate* (should be ~0), *Sacred FP rate* (bounded), *Override frequency*, *Time-to-review*.

F.7 Minimal monitors (pseudo).

```
def sacred_runtime_guard(y, w, S):
  for s in S:
    if s(w,y):
      log_violation(s, w, y)
      return abstain("sacred", s_id=s.id, version=s.version)
  return "pass"
```

(Optional) Appendix G. Auditor Functor Realization (sketch)

G.1 Objects/Morphisms.

- Sys: states; morphisms are tool steps (state, tool) → state'.
- Proofs: statements; morphisms are verified proof transformations.

G.2 Functor A.
Assign each tool step a semantics in Proofs. Require a commuting square:

```
A(state) --ToolProof--> A(state')
  |                       |
 Eval                    Eval
  v                       v
Proof(y)  --------->    Proof(y')
```

If no ToolProof exists, the step cannot justify emission.

Appendix H. STOP Compiler Interface (ABI)

H.1 Pass order.

GrammarGuard → HorizonAnnotator → SelfWitnessFilter

→ GL/Löb Guard → Apophatic/PCI/Authority → STOP_gates → Emission/Receipt

H.2 Return codes.

0 OK; 1 UNSAYABLE; 2 RISK; 3 SACRED; 4 GÖDEL; 5 MODAL; 6 EPISTEMIC

H.3 Telemetry (for audits).

- Feature vector for each gate, thresholds, hashes of tools/compressors, decision trace.

References

Abramsky, Samson, and Achim Jung. 1994. “Domain Theory.” In *Handbook of Logic in Computer Science*, Vol. 3, edited by S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, 1–168. Oxford: Clarendon Press.

Awodey, Steve. 2010. *Category Theory*. 2nd ed. Oxford: Oxford University Press.

Boolos, George. 1993. *The Logic of Provability*. Cambridge: Cambridge University Press.

Bostrom, Nick. 2014. *Superintelligence: Paths, Dangers, Strategies*. Oxford: Oxford University Press.

Caputo, John D. 2006. *The Weakness of God: A Theology of the Event*. Bloomington: Indiana University Press.

Chollet, François. 2019. “On the Measure of Intelligence.” *arXiv* preprint arXiv:1911.01547.

Cover, Thomas M., and Joy A. Thomas. 2006. *Elements of Information Theory*. 2nd ed. Hoboken, NJ: Wiley.

Dionysius the Areopagite (Pseudo-Dionysius). 1987. *The Complete Works*. Translated by Colm Luibheid, with Paul Rorem. New York: Paulist Press.

Gödel, Kurt. 1931. “Über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme I.” *Monatshefte für Mathematik und Physik* 38: 173–198. (English trans. in *Collected Works*, Vol. I.)

Grünwald, Peter D. 2007. *The Minimum Description Length Principle*. Cambridge, MA: MIT Press.

Heidegger, Martin. 1962. *Being and Time*. Translated by John Macquarrie and Edward Robinson. New York: Harper & Row.
———. 1977. “The Question Concerning Technology.” In *Basic Writings*, edited by David Farrell Krell, 287–317. New York: Harper & Row. (Orig. 1954.)

Ibn ʿArabī, Muḥyī al-Dīn. 1980. *The Bezels of Wisdom (Fuṣūṣ al-Ḥikam)*. Translated by R. W. J. Austin. New York: Paulist Press.

Kleene, Stephen C. 1952. *Introduction to Metamathematics*. Amsterdam: North-Holland.

Kolmogorov, Andrey N. 1965. “Three Approaches to the Quantitative Definition of Information.” *Problems of Information Transmission* 1 (1): 1–7.

Kurzweil, Ray. 2005. *The Singularity Is Near: When Humans Transcend Biology*. New York: Viking.

Lawvere, F. William. 1969. “Diagonal Arguments and Cartesian Closed Categories.” In *Category Theory, Homology Theory and Their Applications II*, 134–145. Berlin: Springer.

Li, Ming, and Paul Vitányi. 2008. *An Introduction to Kolmogorov Complexity and Its Applications*. 3rd ed. New York: Springer.

Löb, Martin Hugo. 1955. “Solution of a Problem of Leon Henkin.” *Journal of Symbolic Logic* 20 (2): 115–118.

Mac Lane, Saunders. 1978. *Categories for the Working Mathematician*. 2nd ed. New York: Springer. (1st ed. 1971.)

Maimonides, Moses. 1963–1965. *The Guide of the Perplexed*. 2 vols. Translated by Shlomo Pines. Chicago: University of Chicago Press.

Metzinger, Thomas. 2003. *Being No One: The Self-Model Theory of Subjectivity*. Cambridge, MA: MIT Press.

Rissanen, Jorma. 1978. “Modeling by Shortest Data Description.” *Automatica* 14 (5): 465–471.

Scott, Dana S. 1970. “Outline of a Mathematical Theory of Computation.” In *Proceedings of the 4th Annual Princeton Conference on Information Sciences and Systems*, 169–176.

Shankara (Śaṅkara). 1965. *Brahma-Sutra-Bhāṣya (Commentary on the Brahma-Sūtras).* Translated by Swami Gambhirananda. Calcutta: Advaita Ashrama.

Solovay, Robert M. 1976. “Provability Interpretations of Modal Logic.” *Israel Journal of Mathematics* 25 (3–4): 287–304.

Wittgenstein, Ludwig. 1922. *Tractatus Logico-Philosophicus*. Translated by C. K. Ogden. London: Routledge.
———. 1953. *Philosophical Investigations*. Translated by G. E. M. Anscombe. Oxford: Blackwell.